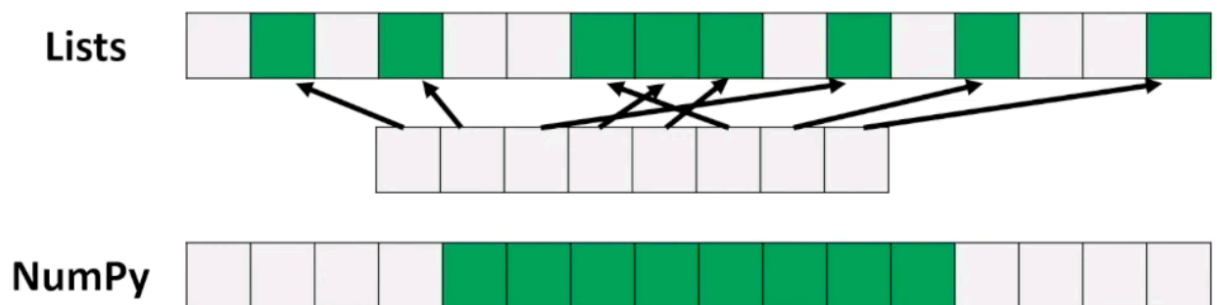


- **the numpy library**
 - -> the package for all scientific computing in Python
 - -> this is an array library
 - -> it can be used to store data in n dimensional arrays
- **this is used over lists**
 - -> lists are slower
 - -> numpy uses fixed types
 - -> the difference between numpy and lists
 - -> 5 is read as 00000101
 - -> this is cast into an int 32 type <- 4 bytes
 - -> numpy reads it as 32 different values, or int16 <- †two bytes
 - -> or into int 8 <- a single byte
 - -> **sizes, reference count, object type and object value**
 - -> the reference count is the amount of times that integer has been pointed at
 - -> you can take the object value -> this is represented as a log
 - -> single integers within lists
 - -> numpy uses less bytes of memory
 - -> **iterating through each item in a numpy array <- we don't have to do type checking each time**
 - -> there is no type checking when iterating through objects
 - -> numpy uses continuous memory
 - -> lists would be scattered around
 - -> memory blocks -> in lists, these aren't necessarily next to each other
 - -> the list contains pointers to the actual information which is scattered around
 - -> this is slower than NumPy
 - -> **numpy uses a contiguous array**
 - -> to store where the total memory is
 - -> in the memory block -> all of the information is in the same place
 - -> this allows for SIMD vector processing
 - -> single instruction multiple data
 - -> to do an addition of multiple values
 - -> performing computations on all the values at the same time
 - -> the cache is also more effectively used
 - -> we can keep the points closer near where we need to access them

Why is NumPy Faster? - Contiguous Memory



Benefits:

- SIMD Vector Processing
- Effective Cache Utilization

- -> longer memories
- -> **lists compared to numpy**
 - -> lists <- insertion, deletion, appending, concatenation
 - -> numpy can do the same thing, but more efficiently
 - -> is we have two different arrays, multiplying them with lists will return an error
 - -> doing this with numpy arrays will times each of the two elements in the lists together
- -> **applications of numpy**
 - -> MATLAB replacement
 - -> it can be used to do maths
 - -> scipy
 - -> functions
 - -> working through the scipy documentation if numpy isn't enough
 - -> for plotting
 - -> backend applications
 - -> pandas, connect 4, digital photography
 - -> you can store images through numpy
 - -> machine learning
 - -> directly and indirectly
 - -> tensors
 - -> these are connected to the tensor library
 - -> the NumPy library
- -> **question**

Why are Numpy arrays faster than regular Python lists?

- options
 - Numpy does not perform type checking while iterating through objects.
 - Numpy uses fixed types.
 - Numpy uses contiguous memory.
 - All of the above. <- This one